

BUS BRIDGE CIRCUIT, BUS CONNECTION SYSTEM, AND DATA ERROR
NOTIFICATION METHOD FOR BUS BRIDGE CIRCUIT

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is based upon and claims the benefit
5 of priority from the prior Japanese Patent Application No.
2003-005284, filed on January 14, 2003, the entire contents
of which are incorporated herein by reference.

BACKGROUND OF THE INVENTION

10

1. Field of the Invention

This invention relates to a bus bridge circuit
connecting two PCI buses or other buses to transfer data
between two buses, a bus connection system, and a data error
15 notification method for the bus bridge circuit, and in
particular relates to a bus bridge circuit which provides
notification of parity errors in data transfers between two
buses, a bus connection system, and a data error
notification method for the bus bridge circuit.

20

2. Description of the Related Art

Various functional devices are connected by buses in
computer systems to realize various functions. With the
spread of personal computers in recent years, a controller
25 and similar are constructed by connecting functional devices
developed for personal computers with a Peripheral Component
Interconnect (PCI) bus.

On the other hand, when devices with different functions are connected by a bus, it is effective to provide bridge circuits in the bus between devices. Fig. 7 is a drawing of the conventional configuration of a bridge circuit (PCI bridge circuit) connected to PCI buses.

The PCI bridge circuit 202 is connected to two PCI buses 300, 400, and performs data transfer between a PCI device 200 connected to the PCI bus 300, and a PCI device 204 connected to the PCI bus 400. The PCI bridge circuit 202 comprises a FIFO (fast-in fast-out) buffer which stores transfer data, and a control circuit which performs control as the target as seen by a PCI device, and also performs control as the master as seen by the other PCI device.

Here, when a read request is issued from the PCI device 200 to the PCI device 204, and the read data is transferred from the PCI device 204 to the PCI device 200 (called a "read operation"), a read request is issued from the PCI device 200 to the PCI bridge circuit 202. Then, the PCI bridge circuit 202 issues a read request to the PCI device 204, and data from the PCI device 204 is pre-fetched to the FIFO buffer via the primary-side PCI bus 400.

On the other hand, the PCB bus protocol provides a CBE (Command Byte Enable) function, so that the validity of data of the PCI bus width (for example, 64 bits = 8 bytes, 32 bits = 4 bytes) can be specified in byte units. That is, the data bits of the PCI bus width can be made valid or invalid in byte units, according to the number of data bits.

required by a PCI device connected to the PCI bus, to acquire the required transfer data from parallel transfer data.

Using this CBE function, for example, in the case of a
5 PCI bus of width 32 bits, a 4-bit CBE signal is issued from the initiator (master) PCI device to control the validity/invalidity of the data in byte units.

Further, in order to protect this CBE signal and the read data (AD signal), during a read operation the XOR
10 (exclusive-OR) of the CBE signal issued from the initiator PCI device to the target PCI device and the bits of all the data outputted by the target PCI device is taken; a parity signal (1 bit) is generated, and the parity signal is sent to the target PCI device.

15 For example, when performing a read operation across the PCI bridge 202 of Fig. 7, the parity for the PCI bus 400 on the target side is generated by XOR'ing the CBE signal issued by the PCI bridge circuit 202 and the read data of the PCI device 204, and the parity for the PCI bus 300 on
20 the initiator side is generated by XOR'ing the CBE signal issued by the PCI device 200 and the read data outputted by the PCI bridge circuit 202.

In the PCI devices 300, 400 on both sides of the PCI bridge circuit 202, the read data has the same value, so
25 that if both CBE signals are the same, the parity will be the same, and it is sufficient for the PCI bridge circuit to transmit to the PCI device 200 the unmodified parity signal

received from the PCI device 204, without generating a parity signal.

However, as stated above, because the CBE signal can be changed, in the prior art the PCI bridge circuit 202
5 generates the parity of the PCI device 200 on the initiator side by XOR'ing the CBE signal issued by the PCI device 200 and the read data outputted by the PCI bridge circuit 202.

On the other hand, when the PCI bridge circuit 202 pre-fetches data from the PCI device 204, the value of the CBE
10 signal on the initiator side cannot be predicted, so that during reading the CBE signals from the PCI bridge circuit 202 are all set to valid (CBE signals are all "0"), to enable accommodation of different patterns of valid bits determined by CBE signals from the initiator-side PCI device
15 200.

That is, there are cases in which the CBE signal values are different on the two sides of the PCI bridge circuit 202. Consequently when the PCI bridge circuit 202 receives a read
20 request from the PCI device 200, if there is a parity error in data pre-fetched from the PCI device 204, the erroneous pre-fetched data and the CBE signal are used to generate the parity, and so there is the problem that the PCI device 200, on receiving this parity signal, cannot recognize the parity error.

25

SUMMARY OF THE INVENTION

Thus, an object of this invention is to provide a bus

bridge circuit, bus connection system, and data error notification method for a bus bridge circuit to notify an initiator-side device of data errors in data pre-fetched by the bus bridge circuit.

5 A further object of this invention is to provide a bus bridge circuit, bus connection system, and data error notification method for a bus bridge circuit to prevent recognition by an initiator-side device of erroneous data as correct data when a data error occurs in data pre-fetched by
10 the bus bridge circuit.

Still another object of this invention is to provide a bus bridge circuit, bus connection system, and data error notification method for a bus bridge circuit, employing a simple configuration, to notify an initiator-side device of
15 data errors in data pre-fetched by the bus bridge circuit.

In order to attain these objects, a bus bridge circuit of this invention issues a read request to a first device in response to a read request from a second device, receives data from the first device via a first bus, and transfers
20 the data to the second device via a second bus, and has a data buffer, which receives and stores the data of the first device, and error detection information generated from the data and from byte enable signals specifying, in units of a prescribed number of bits, the parallel data from the bus
25 bridge circuit on the first bus to be enabled; an error detection information generation circuit, which generates new error detection information from byte enable signals

specifying, in units of a prescribed number of bits, the parallel data from the second device on the second bus to be enabled, and from error detection data received in the data buffer; and a controller, which transfers to the second
5 device via the second bus the data of the data buffer and the new error detection information, according to the byte enable signals of the second device.

A bus connection system of this invention has a second device which issues read requests, a first device which
10 outputs read data in response to the above read requests, and a bus bridge circuit, which is connected to the second device by a second bus and is connected to the first device by a first bus, and transfers read data from the first device via the first bus to the second device via the second
15 bus. And the first device outputs the read data and bus error detection information generated from the read data and from the byte enable signals specifying, in units of a prescribed number of bits the parallel data from the bus bridge circuit on the first bus to be enabled, to the first
20 bus; and the bus bridge circuit has a data buffer which receives and stores the read data from the first bus and the error detection information, an error detection information generation circuit which generates new error detection information from byte enable signals specifying, in units of
25 a prescribed number of bits, the parallel data from the second device on the second bus to be enabled and from error detection information received in the data buffer, and a

controller which transfers to the second device via the second bus the data in the data buffer and the new error detection information, in response to the byte enable signals of the second device.

5 Further, a data error notification method for a bus bridge circuit of this invention issues a read request to a first device in response to a read request from a second device, receives data from the first device via a first bus, and transfers the data to the second device via a second bus, 10 and has a step of receiving and storing in a data buffer, the data of the above first bus and error detection information generated from the data and from byte enable signals specifying, in units of a prescribed number of bits, the parallel data from the bus bridge circuit on the first 15 bus to be enabled; an error detection information generation step of generating new error detection information from byte enable signals specifying, in units of a prescribed number of bits, the parallel data from the above second device on the second bus to be enabled, and from error detection information received in the data buffer; and a step of 20 transferring the data buffer data and the new error detection information to the second device via the second bus, in response to the byte enable signals of the second device.

25 In this invention, new error detection information is generated from error detection information generated by the first device and from byte enable signals specifying, in

units of a prescribed number of bits, the parallel data from the second device on the second bus to be enabled, and the new error detection information is transferred, together with the read data, to the second device, so that even if
5 byte enable values are different on the primary-side and secondary-side buses, data errors (parity errors) on the secondary-side bus can be correctly transmitted to the primary-side bus. Further, this function can be realized easily and at low cost through the addition of a simple
10 circuit.

In this invention, it is preferable that the above controller transmits the byte enable signals, to pre-fetch the data of the first device, to the above first device. As a result, changes in enabled bytes for the second device can
15 be accommodated.

In this invention, it is preferable that the above error detection information generation circuit comprise an XOR circuit to take the XOR of the byte enable signals from the second device and the parity bits received by the above
20 data buffer. As a result, new parity bits can be generated using a simple circuit.

Further in this invention, it is preferable that the controller transmit the byte enable signals in sequence to the first device, in response to ready signals from the
25 first device responding to the read request sent to the first device, and moreover after storing data from the first device in the above data buffer, transmit a ready signal to

the second device, and receives the byte enable signals from the second device. As a result, correct notification of parity is possible without modifying the PCI protocol.

In this invention, it is preferable that the above data
5 buffer comprises a FIFO buffer, so that transmission control, including parity bits and other data errors, is easily performed.

BRIEF DESCRIPTION OF THE DRAWINGS

10 Fig. 1 is a block diagram of a PCI bus connection system of one embodiment of this invention;

Fig. 2 is a drawing of the configuration of a storage system to which the configuration of Fig. 1 is applied;

Fig. 3 is a drawing of the configuration of the PCI
15 bridge circuit of Fig. 1;

Fig. 4 is a timing chart of data transfer operations of the PCI bridge circuit of Fig. 3;

Fig. 5 is a drawing explaining an example of the parity generation processing of Fig. 3;

20 Fig. 6 is a drawing explaining the parity generation operation of Fig. 3; and,

Fig. 7 is a drawing explaining a conventional PCI bridge circuit.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

25 Below, aspects of the invention are explained, in the order PCI bus connection systems, PCI bridge circuits, and

other embodiments.

(PCI Bus Connection Systems)

Fig. 1 is a drawing of the configuration of a PCI bus connection system in one embodiment of this invention; Fig. 2 is a drawing of the configuration of a storage system in which the PCI connection system of Fig. 1 is used as a controller. Fig. 2 shows a RAID (Redundant Arrays of Inexpensive Disks) system using magnetic disks.

As shown in Fig. 2, the storage system comprises a pair of magnetic disk controllers (hereafter called "controllers") 1, 2, and plural magnetic disk devices 50-1 to 50-m and 52-1 to 52-n, connected by lines 11, 12 to this pair of controllers 1, 2.

The controllers 1, 2 are connected, either directly or via network equipment, to a host and server, in a system enabling rapid random-access writing and reading of large amounts of data of the host and server to and from RAID disk drives (magnetic disk devices).

The pair of controllers 1, 2 have the same configuration, comprising as function modules CAs (Channel Adapters) 11, 12, 21, 22, CMs (Centralized Modules) 10, 15 to 19, 20, and 25 to 29, and DAs (Device Adapters) 13, 14, 23, 24.

The CAs (Channel Adapters) 11, 12, 21, 22 are circuits which govern control of the host interface linking the host, and as explained below, comprise fiber channel controllers and similar. The DAs (Device Adapters) 13, 14, 23, 24 are

circuits which exchange commands and data with disk devices in order to control the disk devices 50-1 to 50-m and 52-1 to 52-n, and comprise, for example, fiber channel (FC) circuits, DMA circuits and similar.

5 The CMS (Centralized Modules) have CPUs 10, 20; bridge circuits 17, 27; memory (RAM) 15, 25; compact flash memory 16, 26; IO bridge circuits 18, 28; and pairs of BIOS flash memory 32, 33 and 42, 43.

10 The CMS also have RSPs (Remote Service Processors) 34, 44 and LAN ports for external connection 36, 46. The memory 15, 25 are backed up by batteries, and are used as main memory.

15 The CPUs 10, 20 are connected, via the bridge circuits 17, 27, to the memory 15, 25, the compact flash memory 16, 26, and the IO bridge circuits 18, 28. The memory 15, 25 are used as work area and cache area by the CPUs 10, 20, and the compact flash memory 19, 29 stores the programs to be executed by the CPUs 10, 20. As such programs, a kernel, file access program (read/write program), RAID management
20 program, and similar are stored.

 BIOS flash memory 32, 33, 42, 43 is provided in a paired configuration for redundancy, one used for operation and the other for standby, storing the BIOS. The CPUs 10, 20 execute these programs, and execute read/write processing,
25 RAID management processing and similar.

 The PCI buses 35, 45 are connected via the bridge circuits 17, 27 to the CPUs 10, 20, the compact flash memory

15, 25, the pairs of BIOS flash memory 32, 33, 42, 43, the RSPs 34, 44, and the LAN ports 36, 46.

The RSPs 34, 44 comprise processors which perform various state management and remote services. The LAN ports 36, 46 are used for connection to external LANs (Local Area Networks).

The PCI (Peripheral Component Interconnect) bus 31 connects the CASs 11, 12, 21, 22 and DASs 13, 14, 23, 24, and also connects the CPUs 10, 20 and memory 15, 25 via the IO bridge circuits 18, 28. The PCI bus 31 is also connected to the PCI node link bridge (PNB) circuits 30, 40.

The PCI node link bridge circuit 30 of the controller 1 is connected to the PCI node link bridge circuit 40 of the controller 2, to exchange commands and data between the controllers 1, 2.

The controller 1 controls, for example, the disk devices 50-1 to 50-m, and the controller 2 controls, for example, the disk devices 52-1 to 52-n. In Fig. 2, the disk devices 50-1 to 50-m and 52-1 to 52-n have a RAID5 configuration.

Fig. 1 shows in detail the configuration of the controller 1 only; components which are the same as in Fig. 2 are assigned the same symbols. Controller 2 has the same configuration. In Fig. 1, the CASs 11, 12 each comprise a PCI bridge circuit 7, PCI bus 9, and an FCC (fiber channel controller) 8 which is a PCI device.

That is, the CASs 11, 12 comprise a PCI bridge circuit 7,

which is connected to the IO bridge circuit (PCI device) 18 via the first PCI bus 31; an FCC (PCI device) 8, and a second PCI bus 9 connected to the PCI bridge circuit 7 and FCC 8. The first PCI bus 31 is connected to the above-
5 described PNB 30 and DAS 13, 14.

That is, the input/output system bus connects, through the IO bridge circuit 18, the data processing system (CPU 10, memory 15, 32, 33, 16 and similar) and the input/output system. As described below using Fig. 3, the PCI bridge
10 circuit 7, because a CRC (Cyclic Redundancy Code) is appended to the data of the storage system and a CRC is not appended to data from the host, appends a CRC to data from the host, and deletes the CRC from data for transmission to the host. The bridge circuit 37 is a bridge connecting the
15 flash memory 32, 33 and compact flash memory 16, which are memory units with different properties, to the PCI bus 35.

(PCI Bridge Circuit).

Next, the PCI bridge circuit is explained. Fig. 3 is a block diagram of a PCI bridge circuit; Fig. 4 is a timing
20 chart of data transfer operation via the PCI bridge circuit; Fig. 5 is a drawing explaining an example of parity appending; and Fig. 6 is a drawing explaining the parity generation operation of Fig. 3.

As shown in Fig. 3, the IO bridge circuit (PCI device)
25 18 is connected to the PCI bridge circuit 7 by the first PCI bus 31, and the PCI bridge circuit 7 is connected to the FCC (PCI device) 8 by the second PCI bus 9. A pair of FC

transceivers 80, 82 are provided in the FCC 8. The FC transceivers 80, 82 are connected in an FC (fiber channel) loop, connecting the host and network.

The PCI bridge circuit 7 has an S-P (Secondary-Primary) FIFO (fast-in fast-out) buffer 77 which stores host data (transfer data) of the second PCI bus 9 from the PCI device 8 and transfers the data to the PCI device 18, and a P-S (Primary-Secondary) FIFO buffer 78 which stores the stored data (transfer data) of the first PCI bus 31 from the PCI device 18 and transfers the data to the PCI device 8.

The PCI bridge circuit 7 also has a CRC generation circuit 74, which receives host data of the second PCI bus 9, creates and appends a CRC to the host data, a CRC check circuit 76 which performs a CRC check of stored data for output to the second PCI bus 9 from the P-S FIFO buffer 78 and deletes the CRCs, and a controller 70 which controls the FIFO buffers 77, 78, the CRC generation circuit 74, and the CRC check circuit 76 according to ready signals.

The PCI device 18 has a parity generation circuit 180 which takes the XOR of the 32-bit read data and each of the bits of the CBE signal from the PCI bridge circuit 7, to generate a parity signal for the PCI bus 31. The controller 70 has a parity generation circuit 72 which takes the XOR of the parity signal received from the PCI device 18 and each of the bits of the CBE signal from the PCI device 8, to generate a parity signal for the PCI bus 9.

The operation of the configuration of Fig. 3 is

explained using Fig. 4 through Fig. 6. After the initiator-side PCI device 8 issues a read request to the PCI bridge circuit 7, the PCI bridge circuit 7 performs internal processing conforming to the PCI bus protocol to acquire bus ownership, establishes a data transfer state with the target (complementary)-side PCI device 18, and issues a read request to the PCI device 18.

When preparations for read data transfer are completed, the complementary-side PCI device 18 returns a ready signal to the PCI bridge circuit 7. In response, the PCI bridge circuit 7 transmits a CBE signal (with all bits "0") to the PCI device 18. In response, the PCI device 18 outputs to the PCI bridge circuit 7 32-bit read data 1 and a parity bit P generated by the parity generation circuit 180.

Subsequently, in response to a CBE signal (with all bits "0") from the PCI bridge circuit 7, the PCI device 18 outputs to the PCI bridge circuit 7 the 32-bit read data 2, 3, ..., and parity bits P generated by the parity generation circuit 180.

On the other hand, the PCI bridge circuit 7 stores the output read data and parity bits in the FIFO buffer 78, and after, at least, the 32-bit read data and parity bits are stored, a ready signal is sent to the initiator-side PCI device 8.

As a result, the PCI device 8 transmits a CBE signal (for example, "1000") to the PCI bridge circuit 7. In response, the PCI bridge circuit 7 outputs to the PCI device

8 the 32-bit read data 1 in the FIFO buffer 78, and the parity bits P' generated by the parity generation circuit 72.

Subsequently, in response to CBE signals ("1000") from the PCI device 8, the PCI bridge circuit 7 outputs the 32-bit read data 2, 3, ... and the parity bits P' generated by the parity generation circuit 72 to the PCI device 8.

As shown in Fig. 6, the parity bits from the PCI device 18 are the consequence of an XOR operation on the 32-bit read data and the CBE signals from the PCI bridge circuit 7. That is, if the number of "1"s in the 32-bit read data and the CBE signal is odd, the parity bit P is "1", and if the number of "1"s is even, the parity bit P is "0".

On the other hand, a parity bit P' from the PCI bridge circuit 7 results from an XOR operation on a parity bit P from the PCI device 18 and a CBE signal from the PCI device 8. That is, if the number of "1"s in the parity bit P and the CBE signal is odd, the parity bit P' is "1", and if the number of "1"s is even, the parity bit P' is "0".

That is, as shown in Fig. 5, the CBE signal on the complementary-side bus 31 is "0000", so that the parity bit P is determined by the number of "1"s in the read data. On the other hand, on the initiator-side bus 9, the parity bit P' is determined by the number of "1"s in the parity bit P and the CBE signal.

Hence as shown for the normal case in Fig. 5, when there are no 1-bit errors in the read data, the parity bit P' is "1" if the number of "1"s in the 32-bit read data and

the CBE signal from the PCI device 8 is odd, and "0" if the number of "1"s is even.

Similarly, as shown for the case of a parity error in Fig. 5, when there is a 1-bit error in the read data, the
5 parity bit P' is "1" if the number of "1"s in the 32-bit read data and the CBE signal from the PCI device 8 is odd, and "0" if the number of "1"s is even.

That is, when there is a parity error in the read data from the PCI device 18, the fact that there is a parity
10 error in the read data is reflected in the parity bit P', which is transmitted to the PCI device 8. Similarly, when there is no parity error in the read data from the PCI device 18, the fact that there is no parity error in the read data is reflected in the parity bit P', which is
15 transmitted to the PCI device 8.

In this way, even if the byte enable values are different on the primary-side and secondary-side buses, the parity error on the secondary-side bus 31 can be correctly transmitted to the primary-side bus 9. This function can be
20 easily realized through addition of a simple circuit at low cost.

(Other Embodiments)

In the above-described aspects, RAID storage system controllers with a redundant configuration such as that in
25 Fig. 2 were explained. However, application to PCI bus connection systems for other controller or other data processing devices is possible, and as the physical disks of

the storage system, magnetic disks, optical discs, magneto-optical discs, and various other storage devices can be employed.

Further, the case in which the CBE signal from the PCI device 8 was "1000" was explained, but operation is similar for other bit patterns as well. Also, the case of data transfers in response to read requests from the PCI device 8 to the PCI device 18 was explained, but application to data transfers in response to read requests from the PCI device 18 to the PCI device 8 is also possible; and the PCI device 18 is not limited to an IO bridge circuit, but may be another PCI device. Here, "PCI bus" includes, in addition to a peripheral component interconnect bus, other computer buses.

15 In the above, embodiments of this invention have been explained, but the scope of this invention is not limited thereto; various modifications are possible, and are not excluded from the scope of this invention.

Thus in this invention, new error detection information 20 is generated from the error detection information generated by the first device and from a byte enable signal from the second device, and the new error detection information is transferred together with the read data to the second device, so that even if byte enable values are different for the 25 primary-side and secondary-side buses, secondary-side bus parity errors and other error detection information can be transmitted correctly to the primary-side bus. This

function can be easily realized through the addition of a simple circuit, at low cost.